AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

Claims 1-43 (Cancelled)

44. (Currently Amended) A method comprising:

installing an intermediate representation on a target machine;

generating and running executable code from the intermediate representation;

collecting profile data ~~while the executable code is running, wherein said collecting the profile data comprises~~ including collecting details of a hardware configuration of the target machine and sampling the running of the executable code at a controlled rate;

~~when the CPU is idle,~~ processing the profile data when a CPU of the target machine is idle; and

recompiling ~~software~~ the intermediate representation based on the processed profile data, wherein recompiling comprises using annotations to relate locations in the executable code to the profile data and the intermediate representation.

45. (Previously Presented) The method of claim 44, wherein said processing the profile data comprises generating one or more profiles.

46. (Previously Presented) The method of claim 45, wherein said generating the one or more profiles comprises:

generation of a binary level profile from analysis of the profile data; and derivation of a profile at high level intermediate language from thc binary level profile.

47. (Currently Amended) A computer readable medium having stored thereon data representing instructions that when executed cause a processor to:

generate and run executable code from an intermediate representation installed on a target machine having the processor;

collect profile data ~~while the executable code is running wherein said collecting the profile data comprises~~ including collecting details of a hardware configuration of the target machine and sampling the running of thc executable code ~~at a controlled rate~~;

~~when the CPU is idle,~~ process the profile data; and

recompile ~~software~~ the intermediate representation based on the processed profile data, wherein recompiling comprises using annotations to relate locations in the executable code to the profile data and the intermediate representation.

48. (Previously Presented) The computer readable medium of claim 47, wherein the data further comprises data representing instructions that when excduted cause the processor to:

generate one or more profiles.

49. (Previously Presented) The computer readable medium of claim 47, wherein the data further comprises data representing instructions that when executed cause the processor to:

generate a binary level profile from analysis of the profile data; and derive a profile at high level intermediate language from the binary level profile.

Claims 50-65 (Cancelled)

66. (Currently Amended) A computer system comprising:

a bus;

a communication device coupled with the bus;

a processor coupled with the bus;

a memory coupled with the bus; and

data stored in the memory that represent instructions that when exccuted cause a processor to:

generate and run executable code from an intermediate representation installed on the computer system;

collect profile data ~~while the executable code is running, wherein said collecting the profile data comprises~~ including collecting details of a hardware configuration of the computer system and sampling the running of the executable code at a controlled rate;

~~when the CPU is idle,~~ process the profile data when the processor is idle; and

recompile ~~software~~ thc intermediate representation based on the processed profile data, whcrein recompiling comprises using annotations to relate locations in the executable codc to the profilc data and the intermediate representation.

67. (Previously Presented) The computer system of claim 66, wherein the data further comprises data representing instructions that when executed cause the processor to:

generate one or more profiles.

68. (Previously Presented) The computer system of claim 67, wherein the data further comprises data representing instructions that when executed cause the processor to:

generate a binary level profile from analysis of the profile data; and

derive a profile at high level intermediate language from the binary level profile.

69. (Previously Presented) The computer system of claim 66, wherein the data further comprises data representing instructions that when executed cause the processor to:

create an annotation.

70. (Currently Amended) The method of claim 44, ~~further comprising installing an intermediate representation including generating an initial profile database containing hardware configuration information~~ wherein collecting the details of the hardware configuration of the target machine is performed during installing of the intermediate representation.

71. (Currently Amended) The computer readable medium of claim 47, ~~further comprising installing an intermediate representation including generating an initial profile database containing hardware configuration information~~ wherein collecting the details of the hardware configuration of the target machine is performed during installing of the intermediate representation.

72. (New) The method of claim 44, wherein the hardware configuration includes details of a processor type.

73. (New) The method of claim 44, wherein the hardware configuration includes details of a cache configuration.

74. (New) The method of claim 44, wherein the controlled rate is less than 2%.

75. (New) The computer readable medium of claim 47, wherein the hardware configuration includes details of a processor type.

76. (New) The computer readable medium of claim 47, wherein the rate is less than 2%.

77. (New) The method of claim 44, wherein the controlled rate such that runtime overhead is not noticeable.

78. (New) The method of claim 44, wherein said recompiling is performed by an operating system of the target machine.

79. (New) The method of claim 44, wherein the annotations are implemented as a linked list of data structures.

80. (New) The method of claim 44, wherein an annotation maps a binary level instruction to a source level token.

81. (New) The method of claim 44, wherein an annotation describes how a binary level instruction evolved from a corresponding high level instruction in a prior compilation.

82. (New) The method of claim 44, wherein an annotation includes a compiler optimization phase in which the annotation was created.

83. (New) The method of claim 47, wherein the controlled rate such that runtime overhead is not noticeable.

84. (New) The method of claim 47, wherein said recompiling is performed by an operating system of the target machine.

85. (New) The method of claim 47, wherein the annotations are implemented as a linked list of data structures.